



A theory of computer semiotics par Peter Bøgh Andersen

Jérôme Euzenat

► To cite this version:

Jérôme Euzenat. A theory of computer semiotics par Peter Bøgh Andersen. Bulletin de l'Association Française pour l'Intelligence Artificielle, 2003, 55, pp.55-58. hal-00906621

HAL Id: hal-00906621

<https://inria.hal.science/hal-00906621>

Submitted on 22 Nov 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A theory of computer semiotics

Peter Bøgh Andersen

Cambridge university press, Cambridge (UK), 1997 (0-521-44868-9) première édition 0-521-39366-1, 1990)

Recension par Jérôme Euzenat (INRIA Rhône-Alpes)

Introduction

Pourquoi écrire un compte-rendu sur un livre vieux de plus d'une décennie ? Parce qu'au fur et à mesure de l'augmentation de la communication à travers l'outil informatique, les défauts de prise en compte du sens des « données » manipulées sont de plus en plus apparents. Ils sont aussi vite corrigés que détectés, mais de manière ad hoc, sans vraiment s'intéresser à la construction du sens accomplie par les utilisateurs. Le besoin d'un cadre unificateur se fait sentir et ce livre pourrait en être la base.

Son but est de proposer une théorie de sémiologie informatique, c'est-à-dire de la nature des signes informatiques et de la manière dont les ordinateurs communiquent à l'aide de signes. Il adopte une double perspective par l'approche théorique de son objet et par l'observation, dans le contexte d'utilisation, des impacts de l'informatique. Bien qu'écrit en 1990, et le résultat de travaux de la décennie 80, il n'a, pour l'essentiel pas vieilli (on y reviendra).

Enfin, il s'agit d'un ouvrage de science du langage qui s'intéresse profondément à l'outil informatique, ce n'est pas si fréquent et cela mérite donc toute notre attention.

L'ouvrage est décomposé en trois grandes parties consacrées à la théorie, l'appréhension des ordinateurs comme implémentant des systèmes de signes et l'étude des pratiques et de l'impact de l'ordinateur dans un contexte de travail. Il contient une bibliographie, un index et a été augmenté d'une introduction lors de sa dernière édition.

I. Théorie

La première partie propose les grands principes d'une approche sémiologique des applications informatiques.

Partant d'un cadre structuraliste dans lequel on analyse les productions afin d'isoler les signes et de les interpréter, l'auteur signale à partir de quelques exemples les limites de l'approche. Il propose donc d'étendre le cadre inspiré de Ferdinand de Saussure et de Louis Hjelmslev afin de surmonter ces problèmes. Les extensions portent sur :

- la prise en compte d'actes non symboliques, c'est-à-dire d'actes n'apparaissant pas sur deux plans (signifiant/signifié ou, pour reprendre les termes de l'ouvrage, expression/contenu) et plus spécialement d'actes n'étant pas exprimés mais étant visibles de l'interlocuteur;
- la prise en compte du contexte social, et concernant cette étude la prise en compte de l'environnement de travail et de la tâche à accomplir ;
- l'étude d'un sous-langage (langage de travail) lié à une pratique particulière.

Le premier point est à relier à la théorie des actes de langage. Les deux points suivants étendent le cadre à l'aide des travaux de sociolinguistique, dont le but est de réintroduire le contexte social dans la linguistique, et en particulier ceux de Michael Halliday.

De ces diverses extensions, il appert qu'il est nécessaire d'étendre les méthodes structuralistes comme le test de commutation: remplaçant un élément sur l'un des deux plans — signifiant ou signifié — et observant la modification sur l'autre.

L'auteur ajoute à ces extensions la nécessité de traiter l'évolution du système de signe et de son contexte. Cette préoccupation est à rapprocher de l'aspect social : les rapports sociaux évoluent, les systèmes de signes doivent aussi pouvoir évoluer.

Cette partie ne s'en tient pas à une discussion de salon telle que ce bref compte-rendu pourrait le laisser penser : l'auteur y détaille chacun des concepts utilisés d'un point de vue théorique et à l'aide d'exemples présentés au début du chapitre. Quelques choix peuvent être discutés comme, la restriction à un seul langage de travail au lieu de considérer des sous-langages emboîtés (évoqué cependant), l'affirmation de l'absence de distinction signifiant/signifié en informatique (p142) qui ne tiens pas lorsque l'on s'intéresse à la sémantique des langages (de programmation ou de représentation de connaissance), ou des considérations peu structuralistes sur l'intelligence des systèmes informatiques.

Cependant, cette partie est vraiment passionnante. En particulier, la vision des ordinateurs comme des media n'a fait que se développer depuis l'écriture du livre grâce au développement d'Internet et du Web. Depuis que l'ouvrage a été rédigé, ce rôle de l'ordinateur comme medium a été considérablement renforcé. Mais comme c'est toujours avec son poste de travail que l'on interagit, ce livre garde toute son actualité.

Ma principale critique concerne d'ailleurs les vues proposées dans la section finale de cette première partie (§1.2.5) : «Computers as media». En fait, les ordinateurs devraient être considérés comme des media particuliers : des médias capables de calcul. Il serait donc nécessaire d'étendre la vue proposée par l'auteur en permettant à l'ordinateur la manipulation de signes en tant que signes (on peut considérer que la sémantique des langages informatiques est une première réponse). Une partie de cet aspect, la versatilité peu usuelle des signes informatiques et leur potentiel impact sur le monde, sera prise en compte dans la suite.

II. Ordinateurs

La partie suivante a pour but de proposer une analyse des types de signes utilisés par les ordinateurs. Bien entendu, le plan de l'expression (ou du signifiant) du système de signes, ce qui est analysable, est l'interface entre la machine et l'utilisateur. C'est donc à partir de l'interface que l'analyse peut commencer.

Quels aspects sont particuliers aux signes informatiques ? La question est ambitieuse. Si la plupart des exemples donnés par l'auteur correspondent à l'utilisation d'une interface graphique, la réponse est aussi abstraite que la question. L'auteur caractérise un signe informatique par des caractéristiques permanentes lui permettant d'être identifié, par des caractéristiques temporaires (transcient) lui permettant d'exprimer un état et par sa capacité à être manipulé (handled) par l'utilisateur. Additionnellement, les signes peuvent être utilisés comme des outils, c'est-à-dire manipulés par l'utilisateur afin d'agir sur d'autres signes (par exemple, le vaporisateur du logiciel de dessin utilisé pour colorier une surface ou la clé utilisée pour décrypter un fichier).

Ainsi qu'on le voit, les «Signes informatiques» ne sont pas caractérisés par leur forme (son, ensemble de pixels sur l'écran), mais par des caractéristiques abstraites qui rendent compte des aspects dynamiques et interactifs de l'ordinateur.

Cette grille d'analyse permet à l'auteur de fournir une classification des signes d'interface suivant que le signe peut être manipulé ou non, qu'il montre des caractéristiques temporaires ou non et qu'il peut servir d'outil ou non. Chaque catégorie est illustrée par plusieurs exemples et est décrite par un réseau de pétri définissant les états internes du signe (caractéristiques temporaires), les actions qu'il peut subir (aspect manipulable) ainsi que celles qu'il peut faire subir à d'autres signes (aspect outil). Au passage, ceci fait apparaître des

«**Signes fantômes**» qui n'ont pas de manifestation permanente dans l'interface mais dont l'interaction avec d'autres signes leur font changer certaines caractéristiques temporaires (ainsi le curseur deviendra-t-il un symbole d'interdiction lorsqu'il est placé au-dessus d'un catalogue dans lequel on ne peut écrire). Ces signes fantômes sont déjà un indice de l'activité de l'ordinateur... L'auteur présente de nombreux exemples d'utilisation de ces signes, illustrés par diverses applications (casse-brique, Word, FullPaint, HyperCard et l'application aux chèques postaux suédois qui sert de fil rouge à l'ensemble de l'ouvrage). Au passage, l'auteur a fait une trouvaille astucieuse pour réaliser le test de commutation¹ il le fait non en changeant le programme mais en le remplaçant par une autre version du même programme et en observant les caractéristiques variables et les invariants.

Une fois inventoriés, les différents types de signes sont mis en rapports dans la création de signes composites. Cette dernière partie est moins fouillée que la précédente et mériterait certainement un ouvrage complet à son sujet. Elle est particulièrement importante dans l'interaction avec l'utilisateur et dans l'application de signes outils. Une des difficultés de cette tâche est que les signes révèlent leur signification dans l'interaction (entre eux et avec l'utilisateur) et aucun formalisme n'est disponible pour exprimer cette composition.

Cela révèle une difficulté rencontrée par ailleurs (étudiée par la perspective langage/action et, plus récemment, les systèmes multi-agents)² l'interprétation des signes isolés ne permet pas d'accéder à la totalité du sens. Il est nécessaire d'identifier la discussion dont ils font partie (ce qui, en informatique, est souvent rendu par un protocole).

Alors que l'on pourrait s'en tenir à l'aspect interface, l'une des plus originales contributions de l'ouvrage concerne l'analyse de l'approche de programmation objet. Emboîtant le pas de la métaphore «**Objet**», l'auteur note que la définition d'un objet dans ce cadre est assez proche de celle d'un signe informatique³ une identité, un état, une interface permettant à l'environnement d'interagir avec l'objet et la possibilité d'interagir avec d'autres objets. L'auteur s'est particulièrement intéressé au langage HyperCard dans lequel nombre des objets de base sont des éléments d'interface graphique ce qui en fait le langage de programmation idéal pour la «**Programmation orientée signe**». Le programme qui contrôle le comportement de l'interface agit comme sa sémiologie¹ (on a ici une sémantique opérationnelle de l'interface). Il interprète les différentes actions de l'utilisateur (dans les deux sens du terme). Le langage de programmation est alors sa méta-sémiologie² un langage de description de sémiologie.

Cependant, l'auteur ne va pas beaucoup plus loin dans cette voie. Il considère que «**l'interface est un système de signes manifesté dans les processus [cognitifs] que les personnes créent lorsqu'elles utilisent et interprètent le système**». C'est sans doute oublier qu'un ordinateur ne se contente pas uniquement de manipuler les signes visibles à son interface. L'auteur reconnaît que la fonction «**Rechercher**» d'un gestionnaire de bases de données est complexe et monopolise une large part de l'activité de l'ordinateur. Mais elle n'est que la réponse à une sollicitation de celui-ci. Ce n'est pas le cas de tâche de nettoyage ou de profilage de l'utilisation qu'un ordinateur peut accomplir et qui modifient son comportement. Pour aller plus loin, certains ont ébauché ces dernières années une «**Sémiologie computationnelle**» qui appréhende l'ordinateur non seulement comme communiquant par le biais d'un système de signes mais comme manipulateur de signes voire comme l'instigateur d'un véritable processus de création de signes (sémiosis).

¹ Ici le terme sémiologie est utilisé pour la description d'un système de signes, en l'occurrence celui révélé par l'interaction entre l'utilisateur et l'interface.

Si l'on attribue peu de capacité à l'ordinateur dans l'interprétation des signes, l'auteur ne laisse pas beaucoup plus de place au concepteur. «Un développeur de systèmes informatique est dans la même position que d'autres producteurs de matériaux symboliques, comme des dramaturges : ils peuvent espérer que leurs intentions vont être réalisées lors de la première.». En fait, cette attitude permet de ne pas prendre pour argent comptant les affirmations des manuels d'utilisateurs, mais de justifier les interfaces en contexte comme ce sera fait dans la partie suivante.

Cette partie est celle qui a le plus vieilli, non pas tant dans son fonds, que dans son corpus : rappelez-vous de l'informatique au début des années 1990. Par contre, l'analyse des signes présentés par l'interface a parfaitement résisté au passage du temps.

III. Travail

La dernière partie replace les théories présentées dans les parties précédentes au cœur du contexte de travail. Ceci est analysé dans une application de gestion de la compensation des chèques postaux suédois dans laquelle l'auteur est intervenu pour informatiser l'ensemble du processus (c'est-à-dire remplacer le plus possible des caisses et des chèques physiques circulants de bureau en bureau par une application informatique).

L'idée est d'y montrer comment les principes développés précédemment s'appliquent dans un contexte réel où il est possible de les utiliser pour faire évoluer le système (et il s'agit ici de l'application informatique mais aussi de l'organisation du travail) à la satisfaction de ses utilisateurs. Pour cela l'auteur revient au langage utilisé sur le lieu de travail en analysant conjointement le contexte de travail et le langage utilisé par l'acteur. L'analyse utilise deux techniques : les champs sémantiques permettent d'inventorier les unités syntaxiques du langage utilisé par les intervenants, les jeux de langages permettent d'identifier les conversations permettant d'accomplir un certain travail (affecter une tâche, se coordonner...).

Dans la théorie des champs sémantiques, l'analyse des conversations permet de caractériser les points de vue et d'intérêt des locuteurs. Ils sont classés suivant la tâche à accomplir et la position hiérarchique du locuteur. Une fois le sous-langage (ou le sous-lexique) établi, on montre comment les langages s'accordent ou non au logiciel réalisé, à l'instar de ce que l'on appellerait l'alignement d'ontologies, et comment les utiliser pour concevoir le système (par exemple, pour présenter la «Possession» d'une formule de chèque ce que rendait précédemment inutile leur possession physique). Un des aspects intéressants de ce travail est le fait que l'on ne cherche pas à reproduire coûte que coûte les langages disponibles avant l'arrivée de l'informatique, mais que l'on tienne compte du changement qu'introduit l'irruption de l'informatique dans le service.

L'analyse des conversations permet d'établir des «Jeux de langage» dont le but est de prolonger la théorie des actes de langage par la prise en compte des conversations permettant d'accomplir une certaine tâche et non plus d'actes isolés. Ces jeux de langage s'apparentent donc aux protocoles requis pour interpréter la composition statique et dynamique de signes dont nous déplorions l'absence précédemment. Au passage, l'auteur propose une superbe justification pour une approche externe (structuraliste) des actes de langage : les intentions et croyances des intervenants ne sont pas accessibles aux méthodes d'investigation linguistiques. On cherche donc à isoler les dialogues en terme de «Jeux de langage» et ainsi à construire une typologie des mouvements autorisés (sinon une grammaire des jeux). Bien que ceci aurait pu être très intéressant, l'auteur ne montre pas de jeux de langages remplacés par des interactions avec l'ordinateur. Il évoque par contre les modifications dans ces jeux de

langages après l'informatisation et s'attache longuement à appliquer ces jeux de langages à la conception du système informatique.

Une dernière partie considère la question de savoir qui de l'ordinateur ou de son utilisateur doit avoir le contrôle du processus et, en particulier, du jeu de langage. Andersen considère que les utilisateurs doivent pouvoir contrôler quelle variante de la procédure de travail ils veulent utiliser (que l'interface ne doit pas tourner une variante en norme). Ceci permet à l'organisation du travail d'évoluer. Un argument linguistique pour cela est que la plasticité du matériel est essentielle pour la vie, et donc l'évolution, d'un système de signes.

Cette partie est pour moi la moins aboutie car, si elle applique effectivement les techniques structuralistes à l'analyse de l'organisation du travail, elle ne retourne que très peu au système de signes manipulé par l'ordinateur. Plus souvent les conclusions sont tirées sur des bases sociologiques. C'est plus l'impact de l'informatisation que le système de signes qui est mis en avant. C'est cependant dans la lignée annoncée dès le début par l'auteur : « Les signes et leur vie en société ».

Conclusions

Ce compte-rendu est la lecture d'un informaticien, pas forcément informé mais conscient de ses besoins. Je serais redevable envers quiconque me signalera mes erreurs d'interprétation. Cet ouvrage est point de départ un stimulant et solide. S'il y en avait d'autres, je serais heureux d'en entendre parler.

Pour ma part, je dépasserais la vision naïve de l'ordinateur comme simple médium interactif. Le livre n'indique pas que cette vision introduit un déséquilibre troublant : alors que la plupart des systèmes de signes sont utilisés pour faire communiquer deux entités, la sémiologie informatique présentée ici dote les énoncés d'une certaine autonomie à cause du caractère interactif du médium. Il n'est pas clair que le système de signes serve à la communication entre deux interlocuteurs : l'interactivité semble permettre à l'utilisateur de jouer avec le système de signes plus que de communiquer.

La perspective d'un web sémantique, où les ordinateurs peuvent manipuler la connaissance exprimée, unie les perspectives de l'ordinateur comme médium et l'ordinateur comme « processeur ». Il est alors d'autant plus important de disposer de techniques telles que décrites dans cet ouvrage afin de pouvoir contrôler l'expression de l'ordinateur en fonction de celle de l'humain.

En somme, ce livre est clair et montre une application raisonnée de techniques sémiotiques dans un contexte précis d'utilisation des ordinateurs. C'est un cadre d'analyse et de conception très utile. Il est illustré d'exemples variés et pertinents. Il est parfois un peu répétitif, mais cela permet lors d'une lecture séquentielle de disposer des éléments nécessaires rapidement.

Un aspect rafraîchissant de ce texte est son enracinement nordique. Ainsi, peut-on y rencontrer le cercle de Copenhague en linguistique, l'école Simula de programmation orientée-objet, l'approche participative de la conception, mais aussi les légos, Ibsen et Bergman.

Je le recommanderais à tout chercheur qui se demande quelle sorte de langage est utilisé lors de la communication entre ordinateurs et humains (et entre humains lors de communication médiatisé par ordinateurs) et comment en rendre compte de manière systématique.

Cela dit, le livre est plutôt une présentation de recherche en mouvement qu'un manuel. Il faut le lire pour y retrouver les positions que j'ai soulignées. Ceci est visible à l'absence de réelle conclusion ou perspective.

Il existe d'autres recensions de cet ouvrage disponibles sur le web (demandez-les à votre moteur de recherche favori). Je les ai lues après avoir rédigé celle-ci, ce qui m'a conduit à faire une correction (ne pas utiliser le terme de «registre» utilisé par l'auteur mais celui de sous-langage).